In the Claims:

Claim 1 (currently amended): A computer program product embodied on computer readable media readable by a computing system in a computing environment, for enforcing security policy using style sheet processing, comprising:

computer-readable program code ~~means~~ for obtaining an input document;

computer-readable program code ~~means~~ for obtaining a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element or elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

computer-readable program code ~~means~~ for applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

computer-readable program code ~~means~~ for creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member that is authorized to view that

element to use key distribution material associated with the output document ~~when decrypting~~ to decrypt the encrypted element, and that precludes decryption of the encrypted element by unauthorized community members.

Claim 2 (previously presented): The computer program product according to Claim 1, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 3 (original): The computer program product according to Claim 1, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 4 (original): The computer program product according to Claim 3, wherein said output document is specified in said XML notation.

Claim 5 (currently amended): The computer program product according to Claim 1, wherein said stored policy enforcement objects further comprise computer-readable program code ~~means~~ for overriding a method for evaluating said elements of said input document, and wherein said computer-readable program code ~~means~~ for applying said one or more style sheets further comprises computer-readable program code ~~means~~ for invoking said computer-readable program code ~~means~~ for overriding, thereby causing said markup notation to be added.

Claim 6 (original): The computer program product according to Claim 5, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 7 (currently amended): The computer program product according to Claim 6, wherein said method is a value-of method of said XSL notation, and wherein said computer-readable program code ~~means~~ for overriding said value-of method is by subclassing said value-of method.

Claim 8 (currently amended): The computer program product according to Claim 5, wherein:

said overriding method comprises:

computer-readable program code ~~means~~ for generating said markup notation as encryption tags; and

computer-readable program code ~~means~~ for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said computer-readable program code ~~means~~ for creating said output document further comprises computer-readable program code ~~means~~ for encrypting those elements surrounded by said inserted encryption tags.

Claim 9 (canceled)

Claim 10 (previously presented): The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 11 (previously presented): The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 12 (currently amended): The computer program product according to Claim 1, wherein said computer-readable program code ~~means~~ for creating said output document further comprises:

computer-readable program code ~~means~~ for generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

computer-readable program code ~~means~~ for encrypting said distinct symmetric keys separately for each of said members of said community for which said symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys.

Claim 13 (currently amended): The computer program product according to Claim 12, wherein said computer-readable program code ~~means~~ for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 14 (previously presented): The computer program product according to Claim 1, wherein said encrypted elements in said created output document are encrypted using a cipher block chaining mode encryption process.

Claim 15 (currently amended): The computer program product according to Claim 12, further comprising:

computer-readable program code ~~means~~ for creating a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 16 (currently amended): The computer program product according to Claim 12, further comprising:

computer-readable program code ~~means~~ for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising:

computer-readable program code ~~means~~ for determining zero or more of said communities of which said individual user or process is one of said members;

computer-readable program code ~~means~~ for decrypting, for each of said determined communities, said member-specific version of said symmetric key, thereby creating a decrypted key; and

computer-readable program code ~~means~~ for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities.

Claim 17 (currently amended): The computer program product according to Claim 15, wherein said computer-readable program code ~~means~~ for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising:

computer-readable program code ~~means~~ for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising:

computer-readable program code ~~means~~ for determining zero or more of said key classes which identify said individual user or process as one of said members;

computer-readable program code ~~means~~ for decrypting, for each of said determined key classes, said member-specific version of said encrypted symmetric key, using a private key of said individual user or process, thereby creating a decrypted key; and

computer-readable program code ~~means~~ for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes.

Claim 18 (currently amended): The computer program product according to Claim 16 or Claim 17, further comprising computer-readable program code ~~means~~ for substituting a predetermined text message for any encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 19 (original): The computer program product according to Claim 1, wherein said DTD is replaced by a schema.

Claim 20 (previously presented); The computer program product according to Claim 1, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 21 (original): The computer program product according to Claim 8, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 22 (currently amended): A system for enforcing security policy using style sheet processing in a computing environment, comprising:

an input document;

a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element of elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

means for applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

means for creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member that is authorized to view that element to use key distribution material associated with the output document ~~when decrypting~~ to decrypt the encrypted element, and that precludes decryption of the encrypted element by unauthorized community members.

Claim 23 (previously presented): The system according to Claim 22, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 24 (original): The system according to Claim 22, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 25 (original): The system according to Claim 24, wherein said output document is specified in said XML notation.

Claim 26 (previously presented): The system according to Claim 22, wherein said stored policy enforcement objects further comprise means for overriding a method for evaluating said elements of said input document, and wherein said means for applying said one or more style sheets further comprises means for invoking said means for overriding, thereby causing said markup notation to be added.

Claim 27 (original): The system according to Claim 26, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 28 (original): The system according to Claim 27, wherein said method is a value-of method of said XSL notation, and wherein said means for overriding said value-of method is by subclassing said value-of method.

Claim 29 (previously presented): The system according to Claim 26, wherein:

said overriding method comprises:

means for generating said markup notation as encryption tags; and

means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said means for creating said output document further comprises means for encrypting those elements surrounded by said inserted encryption tags.

Claim 30 (canceled)

Claim 31 (previously presented): The system according to Claim 22, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that visibility policy.

Claim 32 (previously presented): The system according to Claim 22, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that visibility policy.

Claim 33 (previously presented): The system according to Claim 22, wherein said means for creating said output document further comprises:

means for generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

means for encrypting said distinct symmetric keys separately for each of said members of said community for which said symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys.

Claim 34 (previously presented): The system according to Claim 33, wherein said means for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 35 (previously presented): The system according to Claim 22, wherein said encrypted elements in said created output document are encrypted using a cipher block chaining mode encryption process.

Claim 36 (previously presented): The system according to Claim 33, further comprising:

means for creating a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated encrypted elements; (2) an identifier of each member of said unique community;

and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 37 (previously presented): The system according to Claim 33, further comprising:

means for decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising:

means for determining zero or more of said communities of which said individual user or process is one of said members;

means for decrypting, for each of said determined communities, said member-specific version of said symmetric key, thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities.

Claim 38 (previously presented): The system according to Claim 36, wherein said means for encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising:

means for decrypting, for an individual user of process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising:

means for determining zero or more of said key classes which identify said individual user or process as one of said members;

means for decrypting, for each of said determined key classes, said member-specific version of said encrypted symmetric key, using a private key of said individual user or process, thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes.

Claim 39 (previously presented): The system according to Claim 37 or Claim 38, further comprising means for substituting a predetermined text message for encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 40 (original): The system according to Claim 22, wherein said DTD is replaced by a schema.

Claim 41 (previously presented): The system according to Claim 22, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 42 (original): The system according to Claim 29, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

Claim 43 (currently amended): A method for enforcing security policy using style sheet processing in a computing environment, comprising ~~the steps of~~:

providing an input document;

providing a Document Type Definition (DTD) that defines elements of said input document, wherein: (1) an attribute of at least one element defined in said DTD references one of a plurality of stored policy enforcement objects; (2) more than one of said references may reference a single stored policy enforcement object; and (3) each of said stored policy enforcement objects specifies a visibility policy for said referencing element of elements, said visibility policy identifying an encryption requirement for all elements having that visibility policy and a community whose members are authorized to view those elements;

applying one or more style sheets to said input document, thereby adding markup notation to each element of said input document for which said element definition in said DTD references one of said stored policy enforcement objects specifying a visibility policy with a non-null encryption requirement, resulting in creation of an interim transient document that indicates elements of said input document which are to be encrypted; and

creating an output document in which each element of said interim transient document for which markup notation has been added is encrypted in a manner that enables each community member that is authorized to view that element to use key distribution material associated with the output document ~~when decrypting~~ to decrypt the encrypted element, and that precludes decryption of the encrypted element by unauthorized community members.

Claim 44 (previously presented): The method according to Claim 43, wherein said markup notation in said interim transient document comprises tags of a markup language.

Claim 45 (original): The method according to Claim 43, wherein said input document is specified in an Extensible Markup Language (XML) notation.

Claim 46 (original): The method according to Claim 45, wherein said output document is specified in said XML notation.

Claim 47 (currently amended): The method according to Claim 43, wherein said stored policy enforcement objects further comprise executable code for overriding a method for evaluating said elements of said input document, and wherein said applying one or more style sheets to said input document ~~step~~ further comprises overriding said method for evaluating, thereby causing said markup notation to be added.

Claim 48 (original): The method according to Claim 47, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

Claim 49 (currently amended): The method according to Claim 48, wherein said method is a value-of method of said XSL notation, and wherein said ~~step of~~ overriding said value-of method is by subclassing said value-of method.

Claim 50 (currently amended): The method according to Claim 47, wherein:

said ~~step of~~ overriding further comprises ~~the steps of~~:

generating said markup notation as encryption tags; and

inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document for which said visibility policy of said elements in said input document have said non-null encryption requirement; and

said ~~step of~~ creating said output document further comprises ~~the step of~~ encrypting those elements surrounded by said inserted encryption tags.

Claim 51 (canceled)

Claim 52 (previously presented): The method according to Claim 43, wherein said encryption requirement further comprises specification of an encryption algorithm to be used when encrypting elements having that policy.

Claim 53 (previously presented): The method according to Claim 43, wherein said encryption requirement further comprises specification of an encryption algorithm strength value to be used when encrypting elements having that policy.

Claim 54 (currently amended): The method according to Claim 43, wherein said ~~step of~~ creating said output document further comprises ~~the steps of~~:

generating a distinct symmetric key for each unique one of said communities identified by said visibility policy in said stored policy objects for each of said elements of said input document; and

encrypting said distinct symmetric keys separately for each of said members of said community for which said symmetric key was generated, thereby creating member-specific versions of each of said distinct symmetric keys.

Claim 55 (currently amended): The method according to Claim 54, wherein said ~~step of~~ encrypting each of said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions.

Claim 56 (previously presented): The method according to Claim 43, wherein said encrypted elements in said created output document are encrypting using a cipher block chaining mode encryption process.

Claim 57 (currently amended): The method according to Claim 54, further comprising ~~the step of~~:

creating a key class for each of said unique communities, wherein said key class is associated with each of said encrypted elements of said output document for which members of this unique community are authorized viewers, and wherein said key class comprises: (1) an encryption algorithm identifier and key length used when encrypting said associated

encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said member-specific versions of said encrypted symmetric key for each of said identified community members.

Claim 58 (currently amended): The method according to Claim 54, further comprising ~~the step of~~:

decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising ~~the steps of~~:

determining zero or more of said communities of which said individual user or process is one of said members;

decrypting, for each of said determined communities, said member-specific version of said symmetric key, thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities.

Claim 59 (currently amended): The method according to Claim 57, wherein said ~~step of~~ encrypting ~~each of~~ said distinct symmetric keys separately for each of said members uses a public key of said community member as input when creating each of said member-specific versions and further comprising ~~the step of~~:

decrypting, for an individual user or process, only those encrypted elements in said output document for which said individual user or process is one of said authorized community members, further comprising ~~the steps of~~:

determining zero or more of said key classes which identify said individual user or process as one of said members;

decrypting, for each of said determined key classes, said member-specific version of said encrypted symmetric key, using a private key of said individual user or process, thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined key classes.

Claim 60 (currently amended): The method according to Claim 58 or Claim 59, further comprising ~~the step of~~ substituting a predetermined text message for any encrypted elements in said output document which cannot be decrypted for said individual user or process.

Claim 61 (original): The method according to Claim 43, wherein said DTD is replaced by a schema.

Claim 62 (previously presented): The method according to Claim 43, wherein said encryption requirement further comprises specification of an encryption key length.

Claim 63 (original): The method according to Claim 50, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.